

[illegible]

:::info

`confidentialInputs` is a field to store information that should be kept private during computation, and the `data` field is the typical calldata required to interact with a dapp.

If you prefer TypeScript, you can see how to craft your own CCRs in the [examples directory of suave-viem](#).

• • •

• • •

5. Send the Confidential Compute Request

Finally, send the CCR to the network.

```
typescript const res = await wallet.sendTransaction(ccr); console.log(`sent ccr! tx hash: ${res}`);
```

You should see the transaction hash logged to your terminal, like this:

```
bash sent ccr! tx hash: 0xad488fd0a2b428bfa30c7ef8f8ce12e2f7f2554643ad1ca94d15ab11ad5dd9dd
```

Fetching Blockchain Data

To fetch the latest block or transaction receipt, you can use the following functions:

```
typescript async function fetchBlockchainData() { // Get the number of the latest block const latestBlockNumber = await suaveProvider.getBlockNumber();
```

```
console.log('Block number: ', latestBlockNumber);
```

```
// Fetch the latest block const latestBlock = await suaveProvider.getBlock({ blockNumber: latestBlockNumber, includeTransactions: false, });
```

```
console.log('Latest Block:', latestBlock); }
```

```
fetchBlockchainData(); ``
```